Amendments to the Specification

Please replace the paragraph at page 9, line 21, through page 11, line 9, with the following amended paragraph:

Referring back to FIG. 6 and to FIGS. 8A and 8B, redeye detection module 14 identifies a preliminary set of candidate redeye pixels in the redness map 60 (step 82; FIG. 6). In some implementations, the preliminary set of candidate redeye pixels is identified by applying a two-dimensional redness filter to the redness map 60. In one exemplary implementation, the following two-dimensional redness finite impulse response (FIR) filter is applied to the pixel redness measures of the redness map 60:

$$f(x,y) = \begin{cases} 1 & if(|x|< d1) and (|y|< d1) \\ -1 & otherwise \end{cases} \qquad (6)$$

The two-dimensional redness filter is defined with respect to a central kernel pixel area and a pixel area surrounding the kernel pixel area. As shown in FIGS. 8A and 8B, the particular FIR filter implementation of equation (6) is defined with respect to a square kernel area 84 (*AR1*) of side length d1 and a surrounding pixel area 86 (*AR2*) corresponding to a rectangular path defined between a square pixel area of side length d2 and the central kernel pixel area 84, where d1 < d2 (e.g., $d2 = 2 \cdot d1$). In some implementations, the average values of the pixels within the kernel area *AR1* and the surrounding area *AR2* may be computed using integral image processing, where an integral image *S(x, y)* for an input image *I(x, y)* is defined as:

$$S(x,y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i,j) \qquad (7)$$

Given the integral image *S*, the sum of image pixels within an arbitrary rectangle *(x1, x2]* and *(y1, y2]* can be obtained by:

$$Sum(x1, x2, y1, y2) = S(x2, y2) - S(x2, y1) - S(x1, y2) + S(x1, y1) \qquad (8)$$

Based on equation (8), the average value of the pixels within an arbitrary rectangle can be obtained efficiently with three integer additions/subtractions and one division. In the above-described implementation, the average pixel values APV_{R1} and APV_{R2} over areas *AR1* and *AR2*, respectively, are computed and the two-dimensional FIR of equation (6) is applied to the redness map 60 to generate the following redness score (*RS1*) for each corresponding region of the redness map:

$$RS1 = \cancel{AR1 - AR2}\underline{APV_{R1} - APV_{R2}} \quad (9)$$

In another implementation, a nonlinear FIR filter is applied to the redness map 60 to generate the following redness score ($RS2$) for each corresponding region of the redness map:

$$RS2 = APV_{R1} + w \cdot \left(\frac{APV_{R1}}{APV_{R2}} \right)^4 \quad (10)$$

where w is a constant weighting factor, which may be determined empirically. In this equation, APV_{R1} represents the absolute redness of the central kernel square $AR1$, and (APV_{R1}/APV_{R2}) represents the contrast between the central square $AR1$ and the surrounding area $AR2$. The redness score $RS2$ of equation (10) formulates how a red dot region must be sufficiently red while also exhibiting high contrast against its surrounding regions. In the above-described implementations, redeye areas are approximated by square candidate pixel areas. In other embodiments, redeye areas may be approximated by different shapes (e.g., rectangles, circles or ellipses).